

ΕΡΓΑΣΙΑ 1

Γεωργιάδης Μιχάλης, dai17110

Καττίδης Παναγιώτης, dai17200

Κουλαξίδης Γιώργος, dai17177

3.2.2

We didn’t have any mate

# 3.2.4 Πίνακες και Γραφικές Παραστάσεις

Α) Για έναν Node:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **AVG ELAPSED TIME** | **AVG MAP TIME** | **AVG SHUFFLE TIME** | **AVG MERGE TIME** | **AVG REDUCE TIME** |
| REDUCE = 1 | 66.67 | 13 | 31 | 0.33 | 18 |
| REDUCE = 2 | 56.67 | 12 | 28 | 0 | 9.33 |
| REDUCE = 4 | 55.67 | 11 | 27 | 1 | 5.00 |
| REDUCE = 8 | 56 | 11 | 27.67 | 1 | 3.33 |

Ο πίνακας περιλαμβάνει τις τιμές για τους χρόνους που έτρεξαν σε έναν Node.

## 

### Γραφήματα μέσων χρόνων για έναν Node

Συνολική απεικόνιση των πιο πάνω γραφημάτων

Για 3 Nodes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **AVG ELAPSED TIME** | **AVG MAP TIME** | **AVG SHUFFLE TIME** | **AVG MERGE TIME** | **AVG REDUCE TIME** |
| REDUCE = 1 | 58.33 | 15.67 | 16.33 | 1.33 | 18.33 |
| REDUCE = 2 | 47.33 | 15.00 | 13.67 | 1.00 | 10.33 |
| REDUCE = 4 | 44.00 | 15.67 | 14.00 | 1.00 | 5.67 |
| REDUCE = 8 | 45 | 16 | 14.67 | 1 | 4.67 |

Ο πίνακας περιλαμβάνει τις τιμές για τους χρόνους που έτρεξαν σε τρεις Nodes.

### Γραφήματα μέσων χρόνων για τρεις Nodes

Συνολική απεικόνιση των πιο πάνω γραφημάτων

# 3.2.5 Σχολιασμός

### Node 1

Μεταξύ των περιπτώσεων Reduce η κύρια διαφορά για το **AVG ELAPSED TIME** φαίνεται καθώς από τον Reduce = 1 πάμε στο 2, όπου η μέση τιμή μας μεταβαίνει από τα 66,67 sec στα 56,67 sec. Από τον reduce = 2 στο 4 έχουμε διαφορά μόλις 1 sec, δηλαδή από 56,67 sec σε 55,67 sec. Για την διαδικασία με Reduce = 8 το AVG ELAPSED TIME αυξάνεται με την τιμή το φτάνει τα 56 sec.

Όσον αφορά τον **AVG MAP TIME** έχουμε μείωση της τιμής του μέχρι και τον Reduce = 4, για 1, 2 και 4 έχουμε τις τιμές 13, 12 και 11 αντίστοιχα, ενώ για reduce = 8 η τιμή παραμένει στο 11.

Στο **AVG SHUFFLE TIME** η μεγαλύτερη μεταβολή παρατηρείται από 31 σε 28 δευτερόλεπτα για την πρώτη και δεύτερη περίπτωση, μεταξύ του reduce = 2 και reduce = 4 έχουμε μείωση από 28 σε 27, ενώ για reduce = 8 αυξήθηκε στα 27,67 sec.

Για το **AVG MERGE TIME** το reduce = 1 ξεκινάει στα 0.33 sec, με την αύξηση του σε 2 ο μέσος χρόνος πέφτει στα 0 sec. Στις επόμενες δύο περιπτώσεις reduce, δηλαδή ίσο με 4 και 8, ο χρόνος παραμένει σταθερός στο 1 sec.

To **AVG REDUCE TIME** ακολουθεί μια καθοδική πορεία για κάθε περίπτωση αύξησεις του Reduce. Για Reduce = 1 η τιμή του βρίσκεται στα 18 sec, με την αύξηση στα δύο πέφτει σχεδόν στο μισό, δηλαδή 9,33 που αποτελεί και την μεγαλύτερη μείωση, η τιμή 4 μας δίνει μέσο για το **AVG REDUCE TIME** ίσο με 5 sec και τέλος το Reduce = 8 έχει ως αποτέλεσμα την τιμή 3,33 sec.

Ουσιαστικά η μεγαλύτερη διαφορά στους χρόνους παρατηρείται καθώς αυξάνουμε το Reduce από 1 σε 2, από εκεί και πέρα έχουμε ελάχιστη μείωση στην περίπτωση ίσο με 4 και μάλιστα στο **AVG MERGE TIME** υπάρχει αύξηση, ενώ για την περίπτωση ίσο με 8 έχουμε μόνο **AVG REDUCE TIME** να μειώνεται και τα υπόλοιπα να μένουν σταθερά ή να αυξάνονται.

### Node 3

Με την χρήση τριών Node παρατηρείται μια αισθητή μείωση των χρόνων στα δεδομένα μας. Ωστόσο, δεν υπάρχει κάποια ιδιαίτερη διαφορά στο τρόπο με τον οποίο κινήθηκε αυτή η συμπεριφορά του χρόνου μεταξύ των διαφορετικών τιμών του Reduce σε σχέση με την προηγούμενη περίπτωση. Στο **AVG ELAPSED TIME** βλέπουμε ότι εξακολουθεί να υπάρχει η μεγαλύτερη πτώση με την μετάβαση από Reduce =1 σε 2, δηλαδή από 58,33 sec σε 47,33 sec, το Reduce = 4 φτάνει τα 44 sec και στο αυξάνεται στα 45 sec.

Για το **AVG MAP TIME** οι χρόνοι είναι ίσοι για Reduce = 1 και 4, στα 15,67 sec, η μικρότερη τιμή εμφανίζεται στην δεύτερη περίπτωση στα 15 sec και η μεγαλύτερη για 8 στα 16 sec.

Στο **AVG SHUFFLE TIME** η μεγαλύτερη μεταβολή παρατηρείται από 16,33 sec σε 13,67 sec, όπου αποτελεί και την ελάχιστη τιμή, για την πρώτη και δεύτερη περίπτωση, μεταξύ του reduce = 2 και reduce = 4 έχουμε αύξηση στα 14 sec, ενώ για Reduce = 8 αυξήθηκε στα 14,67 sec.

Όσον αφορά **AVG MERGE TIME** το reduce = 1 ξεκινάει στα 1.33 sec και όλες οι υπόλοιπες περιπτώσεις παραμένουν σταθερές ίσες με 1sec.

To **AVG REDUCE TIME** ακολουθεί μια καθοδική πορεία για κάθε περίπτωση αύξησης του Reduce. Για Reduce = 1 η τιμή του βρίσκεται στα 18,33 sec, με την αύξηση σε δύο πέφτει στα 10,,33sec που αποτελεί και την μεγαλύτερη μείωση, η τιμή 4 μας δίνει μέσο για το **AVG REDUCE TIME** ίσο με 5,67 sec και τέλος το Reduce = 8 έχει ως αποτέλεσμα την τιμή 4,67 sec. Αξίζει να σημειωθεί πως η συγκεκριμένη διαδικασία παρουσιάζει μεγαλύτερος χρόνους για 3 Nodes σε σχέση με την αντίστοιχη που έγινε για έναν.

# Περιγραφή κώδικα

## Mapper

Το πρόγραμμα του Mapper αποτελείται από μία εξωτερική δομή ***for*** η οποία σε κάθε επανάληψη με χρήση της εντολής ***sys.stdin*** συνδέεται με ένα input stream το οποίο περιμένει να παραλάβει ένα json-object της μορφής :

{"abstract": "This article applied ...",

"authors": ["Altaf Hossain", "Faisal Zaman", "Mohammed Nasser", "M. Mufakhkharul Islam"],

"n\_citation": 50,

"references": ["2d8c0...", ...],

"title": "Comparison of GARCH.",

"venue": "pattern recognition and machine intelligence",

"year": 2009,

"id": "001c8744-73c4-4b04-9364-22d31a10dbf1"}

Εάν στο object δεν βρεθεί το πεδίο authors τότε γίνεται exception για να προχωρήσει κανονικά η διαδικασία για τις υπόλοιπες εγγραφές. Στη συνέχεια με ένα δεύτερο ***for*** το πρόγραμμα εμφανίζει τον κάθε Author μαζί με τον δείκτη του (δηλαδή τη θέση στην οποία εμφανίζεται σχετικά με τους υπόλοιπους Authors στο συγκεκριμένο object) ακολουθούμενο από ένα άσσο ο οποίος υποδεικνύει το πόσες φορές εμφανίστηκε στη συγκεκριμένη θέση ο συγκεκριμένος Author. (Ο άσσος αυτός λειτουργεί ως αρχικοποίηση και μετέπειτα θα προστεθεί με τους αντίστοιχους άσσους των ιδίων Authors που βρέθηκαν στην ίδια θέση).

## Reducer

Το πρόγραμμα του Reducer παραλαμβάνει με μια επαναληπτική δομή ***for*** μέσω της εντολής ***sys.stdin*** τους Authors ένα-ένα στη μορφή που τους εκτύπωσε ο Mapper ταξινομημένους (με τη βοήθεια του Hadoop) με βάση τα ονόματα των Authors. Για κάθε Author που παραλαμβάνει αφαιρεί με την μέθοδο ***strip()*** τους χαρακτήρες curly-brackets του ***string*** και στη συνέχεια χρησιμοποιεί την μέθοδο ***split()*** για να χωρίσει και να αποθηκεύσει τον author και το count του (μετράει τις φορές που ο συγκεκριμένος author βρέθηκε στη συγκεκριμένη θέση) στις αντίστοιχες μεταβλητές. Στη συνέχεια αν ο ***current\_author*** (ο Author που εξεταζόταν) είναι ίδιος με τον Author που μόλις ανατέθηκε στη μεταβλητή ***author*** τότε η μεταβλητή-μετρητή **current\_count** αυξάνεται κατά ***count***. Εάν ο καινούριος Author δεν είναι ο ίδιος που εξετάστηκε στη προηγούμενη επανάληψη τότε το πρόγραμμα μεταβαίνει στην εντολή ***Else*** και εκτυπώνει το όνομα του Author: ***current\_author*** μαζί με τον μετρητή του: ***current\_count*** και αναθέτει την τιμή ***count*** και ***author*** στις αντίστοιχες μεταβλητές **current\_count** *και* ***current\_author*** για να επαναληφθεί η διαδικασία για νέο Author. Τέλος όσον αφορά το τελευταίο ***for*** το οποίο βρίσκεται έξω από το βρόχο, αν ο τελευταίος Author που ελέγχθηκε δεν εισήλθε στο ***Else*** λόγο του ότι ήταν ίδιος με τον προηγούμενο, το όνομά του μαζί με τον μετρητή δεν εκτυπώθηκε οπότε το πρόγραμμα το εκτυπώνει.